IN THE CLAIMS

Please amend the claims as follows:

1. (Presently Amended) A computer-implemented method of representing data on a computer as a collection of finite distributive lattices for facilitating information retrieval in response to a query, comprising the steps of:

selecting one or more entity types within an application, each entity type being a single entity type;

constructing a ~~first~~ distinct table to represent each ~~an~~ entity type selected in said selecting step, ~~with a column in the table for a respective attribute of the entity type~~ such that there is a one-to-one correspondence between respective tables and entity types;

for each table executing steps of

constructing a column in the table for a respective attribute of the entity type, such that there is a one-to-one correspondence between columns and attributes of the entity type;

constructing a row in the table for each entity of the entity type, such that there is a one-to-one correspondence between rows and entities;

entering attribute data into rows of the ~~first~~ table;

~~constructing~~ specifying a row graph which represents ~~an~~ an externally specified ordering relationship between the rows of the ~~first~~ table; and

~~generating~~ defining a finite distributive lattice (FDL) from the ~~first table~~ row graph to have a member for each row and each distinct ~~combinations~~ combination of the rows of the ~~first~~ table,

wherein two ~~different~~ combinations of ~~members~~ rows are distinct if they do not represent a same ordering relationship.

2. (Presently Amended) The method according to Claim 1, wherein the ~~first~~ table includes one row for each primary entity of the entity type.

3. (Original) The method according to Claim 1, wherein a first entity is included in a second entity if and only if there is a path in the graph from a node corresponding to the first entity to a node corresponding to the second entity.

4. (Presently Amended) The method according to Claim 1, wherein the ordering relationship of the rows of the ~~first~~ table comprises a partially ordered relationship.

5. (Presently Amended) The method according to Claim 1, further comprising the step of:

querying the row graph to determine distinct combinations of the rows of the ~~first~~ table,

wherein two different combinations of members are distinct if they do not represent a same ordering relationship.

6. (Original) The method according to Claim 1, further comprising the step of:

executing a command indicative of an operation $a_1 \leq a_2$ to determine if and only if there is a path in the graph from a_1 to a_2, where a_1 and a_2 are nodes in the row graph.

7. (Original) The method according to Claim 1, further comprising the step of:

executing a command indicative of an operation DOWN(a) in which a resultant table and graph is returned including all rows of the ~~first~~ table and nodes and links of the row graph which are less than or equal to "a", where "a" is a specified node in the row graph.

8. (Original) The method according to Claim 1, wherein the attribute data comprises at least one of simulation data, spatial data, object-orientated data, and relational data.

9. (Original) The method according to Claim 1, further comprising the step of:

executing a command indicative of an operation JOIN(l_1, l_2, ... l_n) to determine a smallest member of the FDL which is greater than or equal to the inputs l_1, l_2, ... l_n.

10. (Original) The method according to Claim 1, further comprising the step of:

executing a command indicative of an operation MEET(l_1, l_2, ... l_n) to determine a largest member of the FDL which is less than or equal to the inputs l_1, l_2, ... l_n.

11. (Original) The method according to Claim 1, wherein any node in the row graph below and connected to another node is included in the other node.

12. (Original) The method according to Claim 1, further comprising the step of:

executing a command indicative of the operation n-ary Cartesian product L1 x L2 x..x Ln, to generate a resultant FDL with columns equal to the disjoint union of the columns of L1, L2, ... Ln, where L1, L2, ... Ln represents any collection of FDLs,

wherein the row-graph is any partial order.

13. (Original) The method according to Claim 1, further comprising the step of:

executing a command indicative of the operation of SELECTION(L, c) to generate a resultant FDL comprising the table rows that satisfy c, the row graph that satisfy c, and all columns of L, where L is a FDL and c is a condition on the rows of L.

14. (Presently Amended) The method according to Claim 1, further comprising the step of:

executing a command indicative of the operation PROJECTION(L, c) to generate a resultant FDL comprising all columns of L that satisfy c, and all table rows and the entire graph of L, where L is a FDL and c is a condition on the columns of L.

15. (Original) The method according to Claim 1, further comprising the step of:

executing a command indicative of the operation UNION(L1, L2) to generate a resultant FDL containing the rows of L1 and the rows of L2, a row graph which is the reflexive, transitive reduction of the union of the row graphs of L1 and L2, and the columns of L1, where L1 and L2 are FDLs with the same columns.

16. (Original) The method according to Claim 1, further comprising the step of:

executing a command indicative of the operation INTERSECTION(L1, L2) to generate a resultant FDL containing those rows of L1 which are the same as rows in L2, a row graph which is the reflexive, transitive reduction of the intersection of the row graphs of L1 and L2, and the columns of L1, where L1 and L2 are FDLs with the same columns.

17. (Presently Amended) The method according to Claim 1, further comprising the step of:

executing a command indicative of RENAME(L, new_name) to change ~~to~~ the name of L to new_name, where L is a FDL and new_name is a name.

18. (Original) The method according to Claim 1, further comprising the step of:

executing a command indicative of the operator CREATE_JIM(L) to create a new row in the table of L and a new node in the row graph of L, where L is a FDL.

19. (Original) The method according to Claim 1, further comprising the step of:

executing a command indicative of DELETE_JIM(L, JIM_ID) to remove a row from the table of L and to remove the corresponding node and any connecting links in the row graph of L, where L is a FDL and JIM_ID identifies a row in L.

20. (Original) The method according to Claim 1, further comprising the step of:

executing a command indicative of the operator CREATE_LINK(L, LESSER_ID, GREATER_ID) to create a link between the node corresponding to LESSER_ID and the node corresponding to GREATER_ID in the row graph of L, where L is a FDL and LESSER_ID and GREATER_ID identify the lesser and greater rows, respectively, of the ordering relationship represented by the link.

21. (Original) The method according to Claim 1, further comprising the step of:

executing a command indicative of the operator DELETE_LINK(L, LESSER_ID, GREATER_ID) to remove a link between the node corresponding to LESSER_ID and the node corresponding to GREATER_ID in the row graph of L, where L is a FDL and LESSER_ID and GREATER_ID identify the lesser and greater rows, respectively, of the ordering relationship represented by the link.

22. (Presently Amended) A computer implemented data model for facilitating information retrieval via a query, comprising:

a ~~first~~ first table and other tables respectively configured to represent ~~an~~ a single entity type from a group of entity types such that there is a one-to-one correspondence between respective tables and entity types, said first table having

a column ~~in the table~~ for a respective attribute of the entity type such that there is a one-to-one correspondence between columns and respective attributes of the entity type, and ~~having rows~~

a different row for each entity of the entity types such that there is a one-to-one correspondence between rows and respective entities, wherein each row is externally specified and includes ~~entered with~~ attribute data for an entity that corresponds with the row; ~~and~~

a row graph which represents an ordering relationship between the rows of the first table; and

a finite distributive lattice (FDL) ~~generated~~ defined from the first table to have distinct combinations of the rows of the first table, wherein two different combinations of members are distinct if they do not represent a same ordering relationship.

23. (Original) The data model according to Claim 22, wherein the first table includes one row for each primary entity of the entity type.

24. (Presently amended) The ~~date~~ data model according to Claim 22, wherein a first entity is included in a second entity if and only if there is a path in the graph from a node corresponding to the first entity to a node corresponding to the second entity.

25. (Original) The data model according to Claim 22, wherein the ordering relationship of the rows of the first table comprises a partially ordered relationship.

26. (Original) The data model according to Claim 22, wherein distinct combination of the rows of the first table are determined by querying the row graph, and

wherein two different combinations of members are distinct if they do not represent a same ordering relationship.

27. (Original) The data model according to Claim 22, further comprising:

a command indicative of an operation $a_1 \leq a_2$ to determine if and only if there is a path in the graph from a_1 to a_2, where a_1 and a_2 are nodes in the row graph.

28. (Original) The data model according to Claim 22, further comprising:

a command indicative of an operation DOWN(a) in which a resultant table and graph is returned including all rows of the first table and nodes and links of the row graph which are less than or equal to "a", where "a" is a specified node in the row graph.

29. (Original) The data model according to Claim 22, wherein the attribute data comprises at least one of simulation data, spatial data, object-orientated data, and relational data.

30. (Original) The data model according to Claim 22, further comprising:

a command indicative of an operation JOIN(l_1, l_2, ... l_n) to determine a smallest member of the FDL which is greater than or equal to the inputs l_1, l_2, ... l_n.

31. (Original) The data model according to Claim 22, further comprising:

a command indicative of an operation MEET(l_1, l_2, ... l_n) to determine a largest member of the FDL which is less than or equal to the inputs l_1, l_2, ... l_n.

32. (Original) The data model according to Claim 22, wherein any node in the row graph below and connected to another node is included in the other node.

33. (Original) The data model according to Claim 22, further comprising:

a command indicative of the operation n-ary Cartesian product L1 x L2 x. . x Ln, to generate a resultant FDL with columns equal to the disjoint union of the columns of L1, L2, . . . Ln, where L1, L2, . . . Ln represents any collection of FDLs,

wherein the row-graph is any partial order.

34. (Original) The data model according to Claim 22, further comprising:

a command indicative of the operation of SELECTION(L, c) to generate a resultant FDL comprising the table rows of L that satisfy c, the row graph that satisfy c, and all columns of L, where L is a FDL and c is a condition on the rows of L.

35. (Presently Amended) The data model according to Claim 22, further comprising:

a command indicative of the operation PROJECTION(L, c) to generate a resultant FDL comprising all columns of L that satisfy c, and all table rows and the entire graph of L, where L is a FDL and c is a condition on the columns of L.

36. (Original) The data model according to Claim 22, further comprising:

a command indicative of the operation UNION(L1, L2) to generate a resultant FDL containing the rows of L1 and the rows of L2, a row graph which is the reflexive, transitive reduction of the union of the row graphs if L1 and L2, and columns of L1, where L1 and L2 are FDLs with the same columns.

37. (Original) The data model according to Claim 22, further comprising:

a command indicative of the operation INTERSECTION(L1, L2) to generate a resultant FDL containing those rows of L1 which are the same as rows in L2, a row graph which is the reflexive, transitive reduction of the intersection of the row graphs of L1 and L2, and the columns of L1, where L1 and L2 are FDLs with the same columns.

38. (Original) The data model according to Claim 22, further comprising:

a command indicative of RENAME(L, new_name) to change ~~to~~ the name of L to new_name, where L is a FDL and new_name is a name.

39. (Original) The data model according to Claim 22, further comprising:

a command indicative of the operator CREATE_JIM(L) to create a new row in the table of L and a new node in the row graph of L, where L is a FDL.

40. (Original) The data model according to Claim 22, further comprising:

a command indicative of DELETE_JIM(L, JIM_ID) to remove a row from the table of L and to remove the corresponding node and any connecting links in the row graph of L, where L is a FDL and JIM_ID identifies a row in L.

41. (Original) The data model according to Claim 22, further comprising:

a command indicative of the operator CREATE_LINK(L, LESSER_ID, GREATER_ID) to create a link between the node corresponding to LESSER_ID and the node corresponding to GREATER_ID in the row graph of L, where L is a FDL and LESSER_ID and GREATER_ID identify the lesser and greater rows, respectively, of the ordering relationship represented by the link.

42. (Original) The data model according to Claim 22, further comprising:

a command indicative of the operator DELETE_LINK(L, LESSER_ID, GREATER_ID) to remove a link between the node corresponding to LESSER_ID and the node corresponding to GREATER_ID in the row graph of L, where L is a FDL and LESSER_ID and GREATER_ID identify the lesser and greater rows, respectively, of the ordering relationship represented by the link.

43. (Presently Amended) A computer-implemented method of representing data on a computer as a collection of finite sheaves for facilitating information retrieval in response to a query, comprising the steps of:

selecting one or more entity types within an application, each entity type being a single entity type;

constructing a ~~first~~ distinct table to represent ~~each~~ an entity type selected in said selecting step, such that there is a one-to-one correspondence between respective tables and entity types;

for each table executing steps of:

selecting another table, hereafter referred to as the schema table of the current table;

constructing a column in the current table for each row of the respective schema table, such that there is a one-to-one correspondence between columns in the current table and rows in the schema table;

interpreting each column of the table as an attribute of the associated entity type;

interpreting a row graph of the schema table as a column graph of the current table, said column graph representing an ordering relationship between the columns of the current table;

constructing a row in the current table for each entity of the associated entity type, such that there is a one-to-one correspondence between rows and entities;

entering attribute data into rows of the current table;

specifying a row graph which represents an externally specified ordering relationship between the rows of the current table;

defining a finite distributive lattice (FDL) from the row graph to have a member for each row and each distinct combination of the rows of the current table, wherein two combinations of rows are distinct if they do not represent a same ordering relationship; and

defining a finite sheaf from the table, row graph and column graph

~~with a column in the table for a respective attribute of the entity type;~~

~~entering attribute data into rows of the first tableentities;~~

~~constructing specifying a row graph which represents an externally specified ordering relationship between the rows of the first table;~~

~~generating defining a finite distributive lattice (FDL) from the first table row graph to have a member for each row and each distinct combinations combination of the rows of the~~

~~first table, wherein two different combinations of members rows are distinct if they do not represent a same ordering relationship;~~

~~assigning a column graph which represents an ordering relationship between columns of the first table, the column graph being a row graph from a second table; and~~

~~interpreting the table, row graph and column graph as a finite sheaf~~.

Claim 44 (Cancelled)

Claim 45 (Original): The method according to Claim 43, wherein a first entity is included in a second entity if and only if there is a path in the graph from a node corresponding to the first entity to a node corresponding to the second entity.

Claim 46 (Original): The method according to Claim 43, wherein the ordering relationship of the rows of the first table comprises a partially ordered relationship.

Claim 47 (Original): The method according to Claim 43, further comprising the step of:

querying the row graph to determine distinct combinations of the rows of the first table,

wherein two different combinations of members are distinct if they do not represent a same ordering relationship.

Claim 48 (Original): The method according to Claim 43, further comprising the step of:

executing a command indicative of an operation a1<a2 to determine if and only if there is a path in the graph from a1 to a2, where a1 and a2 are nodes in the row graph.

Claim 49 (Original): The method according to Claim 43, further comprising the step of:

executing a command indicative of an operation DOWN(a) in which a resultant table and graph is returned including all rows of the first table and nodes and links of the row graph which are less than or equal to "a", where "a" is a specified node in the row graph.

Claim 50 (Original): The method according to Claim 43, wherein the attribute data comprises at least one of simulation data, spatial data, object-orientated data, and relational data.

Claim 51 (Original): The method according to Claim 43, further comprising the step of:

executing a command indicative of an operation JOIN(l1, l2, ... ln) to determine a smallest member of the sheaf which is greater than or equal to the inputs l1, l2, ... ln.

Claim 52 (Original): The method according to Claim 43, further comprising the step of:

executing a command indicative of an operation MEET(l1, l2, ... ln) to determine a largest member of the sheaf which is less than or equal to the inputs l1, l2, ... ln.

Claim 53 (Original): The method according to Claim 43, further comprising the step of:

executing a command indicative of an operation EXP(L) to generate a resultant sheaf with a column graph corresponding to the row graph of L, where L is a sheaf.

Claim 54 (Presently Amended): The method according to Claim 43, further comprising the step of:

executing a command indicative of an operation LOG(L) to determine the schema ~~lattice~~ sheaf of ~~the sheaf~~ L, where L is ~~the~~ a sheaf.

Claim 55 (Presently Amended): The method according to Claim 43, further comprising the step of:

executing a command indicative of an operation RESTRICT L TO S to determine a projection of the sheaf onto columns in the down set of S, where S is a member of the schema of L and L is ~~the~~ a sheaf.

Claim 56 (Original): The method according to Claim 43, wherein any node in the row graph below and connected to another node is included in the other node.

Claim 57 (Presently Amended): The method according to Claim 43, further comprising the step of:

executing a command indicative of the operation n-ary Cartesian product L1 x L2 x..x Ln, to generate a resultant sheaf with a schema ~~lattice~~ sheaf equal to the union of the schema of L1, L2, ... Ln, where L1, L2, ... Ln represents any collection of sheaves.

Claim 58 (Presently Amended): The method according to Claim 43, further comprising the step of:

executing a command indicative of the operation of SELECTION(L, c) to generate a resultant sheaf comprising the table rows that satisfy c, the row graph that satisfy c, and the

entire column graph and all columns of L, where L is ~~an~~ a sheaf and c is a condition on the rows of L.

Claim 59 (Original): The method according to Claim 43, further comprising the step of:

executing a command indicative of the operation PROJECTION(L, c) to generate a resultant sheaf comprising the column graph and columns of L that satisfy c, and all table rows, and the entire row graph of L, where L is a sheaf and c is a condition on the columns of L.

Claim 60 (Original): The method according to Claim 43, further comprising the step of:

executing a command indicative of the operation UNION(L1, L2) to generate a resultant sheaf containing the rows of L1 and the rows of L2, a row graph which is the reflexive, transitive reduction of the union of the row graphs of L2 and L2, and the column graph of L1, where L1 and L2 are sheaves with the same column graph.

Claim 61 (Original): The method according to Claim 43, further comprising the step of:

executing a command indicative of the operation INTERSECTION(L1, L2) to generate a resultant sheaf containing those rows of L1 which are the same as rows in L2, a row graph which is the reflexive, transitive reduction of the intersection of the row graphs of L1 and L2, and the column graph of L1, where L1 and L2 are sheaves with the same column graph.

Claim 62 (Presently Amended): The method according to Claim 43, further comprising the step of:

executing a command indicative of RENAME(L, new_name) to change ~~to~~ the name of L to new_name, where L is a sheaf and new_name is a name.

Claim 63 (Original): The method according to Claim 43, further comprising the step of:

executing a command indicative of the operator CREATE_JIM(L) to create a new row in the table of L and a new node in the row graph of L, where L is a sheaf.

Claim 64 (Original): The method according to Claim 43, further comprising the step of:

executing a command indicative of DELETE_JIM(L, JIM_ID) to remove a row from the table of L and to remove the corresponding node and any connecting links in the row graph of L, where L is a sheaf and JIM_ID identifies a row in L.

Claim 65 (Original): The method according to Claim 43, further comprising the step of:

executing a command indicative of the operator CREATE_LINK(L, LESSER_ID, GREATER_ID) to create a link between the node corresponding to LESSER_ID and the node corresponding to GREATER_ID in the row graph of L, where L is a sheaf and LESSER_ID and GREATER_ID identify the lesser and greater rows, respectively, of the ordering relationship represented by the link.

Claim 66 (Original): The method according to Claim 43, further comprising the step of:

executing a command indicative of the operator DELETE_LINK(L, LESSER_ID, GREATER_ID) to remove a link between the node corresponding to LESSER_ID and the node corresponding to GREATER_ID in the row graph of L, where L is a sheaf and LESSER_ID and GREATER_ID identify the lesser and greater rows, respectively, of the ordering relationship represented by the link.

67. (Presently Amended) A data model encoded on a computer-readable medium as a collection of finite sheaves for facilitating information retrieval in response to a query, comprising:

a first table and other tables respectively configured to represent a single entity type from a group of entity types, such that there is a one-to-one correspondence between respective tables and entity types, said first table having:

an association with a second table, hereafter referred to as the schema table of the first table;

a column for each row of the respective schema table, such that there is a one-to-one correspondence between columns in the first table and rows in the schema table; each said column corresponding to an attribute of the entity type associated with the first table;

a column graph defined to be a row graph of the respective schema table, said column graph representing an ordering relationship between the columns of the current table;

a row for each entity of the associated entity type, such that there is a one-to-one correspondence between rows and entities;

attribute data included in the rows;

a row graph which represents an externally specified ordering relationship between the rows of the first table;

a finite distributive lattice (FDL) defined from the row graph to have a member for each row and each distinct combination of the rows of the first table, wherein two combinations of rows are distinct if they do not represent a same ordering relationship; and

a finite sheaf corresponding to the table, row graph and column graph

~~to represent an entity type having a column in the table for a respective attribute of the entity type, and having rows entered with attribute data;~~

~~a row graph which represents an ordering relationship between the rows of the first table;~~

~~a finite distributive lattice (FDL) generated from the first table to have distinct combinations of the rows of the first table, wherein two different combinations of members rows are distinct if they do not represent a same ordering relationship;~~

~~a column graph which represents an ordering relationship between columns of the first table, the column graph being a row graph from a second table; and~~

~~a sheaf corresponding to the table, row graph and column graph~~.

Claim 68 (Cancelled)

Claim 69 (Presently amended): The ~~date~~ data model according to Claim 67, wherein a first entity is included in a second entity if and only if there is a path in the graph from a node corresponding to the first entity to a node corresponding to the second entity.

Claim 70 (Original): The data model according to Claim 67, wherein the ordering relationship of the rows of the first table comprises a partially ordered relationship.

Claim 71 (Original): The data model according to Claim 67, wherein distinct combination of the rows of the first table are determined by querying the row graph, and

wherein two different combinations of members are distinct if they do not represent a same ordering relationship.

Claim 72 (Original): The data model according to Claim 67, further comprising:

a command indicative of an operation a1<a2 to determine if and only if there is a path in the graph from a1 to a2, where a1 and a2 are nodes in the row graph.

Claim 73 (Original): The data model according to Claim 67, further comprising:

a command indicative of an operation DOWN(a) in which a resultant table and graph is returned including all rows of the first table and nodes and links of the row graph which are less than or equal to "a", where "a" is a specified node in the row graph.

Claim 74 (Original): The data model according to Claim 67, wherein the attribute data comprises at least one of simulation data, spatial data, object-orientated data, and relational data.

Claim 75 (Original): The data model according to Claim 67, further comprising:

a command indicative of an operation JOIN(l1, l2, ... ln) to determine a smallest member of the sheaf which is greater than or equal to the inputs l1, l2, ... ln.

Claim 76 (Original): The data model according to Claim 67, further comprising:

a command indicative of an operation MEET(l1, l2, ... ln) to determine a largest member of the sheaf which is less than or equal to the inputs l1, l2, ... ln.

Claim 77 (Original): The data model according to Claim 67, further comprising:

a command indicative of an operation EXP(L) to generate a resultant sheaf with a column graph corresponding to the row graph of L, where L is a sheaf.

Claim 78 (Original): The data model according to Claim 67, further comprising:

a command indicative of an operation LOG(L) to determine the schema ~~lattice of the~~ sheaf of L, where L is ~~the~~ a sheaf.

Claim 79 (Presently Amended): The data model according to Claim 67, further comprising:

a command indicative of an operation RESTRICT L TO S to determine a projection of ~~the sheaf~~ L onto columns in the down set of S, where S is a member of the schema of L and L is ~~the~~ a sheaf.

Claim 80 (Original): The data model according to Claim 67, wherein any node in the row graph below and connected to another node is included in the other node.

Claim 81 (Presently Amended): The data model according to Claim 67, further comprising:

a command indicative of the operation n-ary Cartesian product L1 x L2 x..x Ln, to generate a resultant sheaf with a schema ~~lattice~~ sheaf equal to the union of the schema of L1, L2, ... Ln, where L1, L2, ... Ln represents any collection of sheaves.

Claim 82 (Original): The data model according to Claim 67, further comprising:

a command indicative of the operation of SELECTION(L, c) to generate a resultant sheaf comprising the table rows that satisfy c, the row graph that satisfy c, and the entire column graph and all columns of L, where L is a sheaf and c is a condition on the rows of L.

Claim 83 (Original): The data model according to Claim 67, further comprising:

a command indicative of the operation PROJECTION(L, c) to generate a resultant sheaf comprising the column graph and columns of L that satisfy c, and all table rows, and the entire row graph of L, where L is a sheaf and c is a condition on the columns of L.

Claim 84 (Original): The data model according to Claim 67, further comprising:

a command indicative of the operation UNION(L1, L2) to generate a resultant sheaf containing the rows of L1 and the rows of L2, a row graph which is the reflexive, transitive reduction of the union of the row graphs of L1 and L2, where L1 and L2 are sheaves with the same column graph.

Claim 85 (Original): The data model according to Claim 67, further comprising:

a command indicative of the operation INTERSECTION(L1, L2) to generate a resultant sheaf containing those rows of L1 which are the same as rows in L2, a row graph which is the reflexive, transitive reduction of the intersection of the row graphs of L1 and L2, and the column graph of L1, where L1 and L2 are sheaves with the same column graph.

Claim 86 (Original): The data model according to Claim 67, further comprising:

a command indicative of RENAME(L, new_name) to change ~~to~~ the name of L to new_name, where L is a sheaf and new_name is a name.

Claim 87 (Original): The data model according to Claim 67, further comprising:

a command indicative of the operator CREATE_JIM(L) to create a new row in the table of L and a new node in the row graph of L, where L is a sheaf.

Claim 88 (Original): The data model according to Claim 67, further comprising:

a command indicative of DELETE_JIM(L, JIM_ID) to remove a row from the table of L and to remove the corresponding node and any connecting links in the row graph of L, where L is a sheaf and JIM_ID identifies a row in L.

Claim 89 (Original): The data model according to Claim 67, further comprising:

a command indicative of the operator CREATE_LINK(L, LESSER_ID, GREATER_ID) to create a link between the node corresponding to LESSER_ID and the node corresponding to GREATER_ID in the row graph of L, where L is a sheaf and LESSER_ID and GREATER_ID identify the lesser and greater rows, respectively, of the ordering relationship represented by the link.

Claim 90 (Original): The data model according to Claim 67, further comprising:

a command indicative of the operator DELETE_LINK(L, LESSER_ID, GREATER_ID) to remove a link between the node corresponding to LESSER_ID and the node corresponding to GREATER_ID in the row graph of L, where L is a sheaf and LESSER_ID and GREATER_ID identify the lesser and greater rows, respectively, of the ordering relationship represented by the link.

IN THE DRAWINGS

The attached sheet of drawings includes changes to Fig. 4C. This sheet, which includes Fig. 4C, replaces the original sheet including Fig. 4C.

Attachment: Replacement Sheet